In [107]:

```
import pandas as pd
```

In [108]:

```
matches = pd.read_csv('matches.csv')
```

In [109]:

```
matches.head()
```

Out[109]:

	Unnamed: 0	date	time	comp	round	day	venue	result	gf	ga	...	match report	notes	sh	sot	dist	fk	pk	pkatt	seas
0	1	2021-08-15	16:30	Premier League	Matchweek 1	Sun	Away	L	0.0	1.0	...	Match Report	NaN	18.0	4.0	16.9	1.0	0.0	0.0	20
1	2	2021-08-21	15:00	Premier League	Matchweek 2	Sat	Home	W	5.0	0.0	...	Match Report	NaN	16.0	4.0	17.3	1.0	0.0	0.0	20
2	3	2021-08-28	12:30	Premier League	Matchweek 3	Sat	Home	W	5.0	0.0	...	Match Report	NaN	25.0	10.0	14.3	0.0	0.0	0.0	20
3	4	2021-09-11	15:00	Premier League	Matchweek 4	Sat	Away	W	1.0	0.0	...	Match Report	NaN	25.0	8.0	14.0	0.0	0.0	0.0	20
4	6	2021-09-18	15:00	Premier League	Matchweek 5	Sat	Home	D	0.0	0.0	...	Match Report	NaN	16.0	1.0	15.7	1.0	0.0	0.0	20

5 rows × 28 columns

In [110]:

```
matches.columns
```

Out[110]:

```
Index(['Unnamed: 0', 'date', 'time', 'comp', 'round', 'day', 'venue', 'result',
       'gf', 'ga', 'opponent', 'xg', 'xga', 'poss', 'attendance', 'captain',
       'formation', 'referee', 'match report', 'notes', 'sh', 'sot', 'dist',
       'fk', 'pk', 'pkatt', 'season', 'team'],
      dtype='object')
```

In [111]:

```
matches.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1389 entries, 0 to 1388
Data columns (total 28 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Unnamed: 0    1389 non-null   int64  
 1   date          1389 non-null   object 
 2   time          1389 non-null   object 
 3   comp          1389 non-null   object 
 4   round         1389 non-null   object 
 5   day           1389 non-null   object 
 6   venue         1389 non-null   object 
 7   result        1389 non-null   object 
 8   gf            1389 non-null   float64
 9   ga            1389 non-null   float64
 10  opponent      1389 non-null   object 
 11  xg            1389 non-null   float64
 12  xga           1389 non-null   float64
 13  poss          1389 non-null   float64
 14  attendance    693 non-null    float64
 15  captain       1389 non-null   object 
 16  formation     1389 non-null   object 
 17  referee       1389 non-null   object 
 18  match report  1389 non-null   object 
 19  notes         0 non-null      float64
 20  sh            1389 non-null   float64
 21  sot           1389 non-null   float64
 22  dist          1388 non-null   float64
 23  fk            1389 non-null   float64
 24  pk            1389 non-null   float64
 25  pkatt         1389 non-null   float64
 26  season        1389 non-null   int64  
 27  team          1389 non-null   object 
dtypes: float64(13), int64(2), object(13)
memory usage: 304.0+ KB
```

In [112]:

```
len(matches)
```

Out[112]:

```
1389
```

In [113]:

```
len(matches.columns)
```

Out[113]:

```
28
```

In [114]:

```
matches.isna().sum()
```

Out[114]:

```
Unnamed: 0         0
date               0
time               0
comp               0
round              0
day                0
venue              0
result             0
gf                 0
ga                 0
opponent           0
xg                 0
xga                0
poss               0
attendance       696
captain            0
formation          0
referee            0
match report       0
notes           1389
sh                 0
sot                0
dist               1
fk                 0
pk                 0
pkatt              0
season             0
team               0
dtype: int64
```

In [115]:

```
matches = matches.drop(['notes', 'attendance'], axis=1)
```

In [116]:

```
matches.head()
```

Out[116]:

	Unnamed: 0	date	time	comp	round	day	venue	result	gf	ga	...	referee	match report	sh	sot	dist	fk	pk	pkatt	s
0	1	2021-08-15	16:30	Premier League	Matchweek 1	Sun	Away	L	0.0	1.0	...	Anthony Taylor	Match Report	18.0	4.0	16.9	1.0	0.0	0.0	
1	2	2021-08-21	15:00	Premier League	Matchweek 2	Sat	Home	W	5.0	0.0	...	Graham Scott	Match Report	16.0	4.0	17.3	1.0	0.0	0.0	
2	3	2021-08-28	12:30	Premier League	Matchweek 3	Sat	Home	W	5.0	0.0	...	Martin Atkinson	Match Report	25.0	10.0	14.3	0.0	0.0	0.0	
3	4	2021-09-11	15:00	Premier League	Matchweek 4	Sat	Away	W	1.0	0.0	...	Paul Tierney	Match Report	25.0	8.0	14.0	0.0	0.0	0.0	
4	6	2021-09-18	15:00	Premier League	Matchweek 5	Sat	Home	D	0.0	0.0	...	Jonathan Moss	Match Report	16.0	1.0	15.7	1.0	0.0	0.0	

5 rows × 26 columns

In [117]:

```
matches= matches.drop('Unnamed: 0', axis=1)
```

In [118]:

```
matches.head()
```

Out[118]:

	date	time	comp	round	day	venue	result	gf	ga	opponent	...	referee	match report	sh	sot	dist	fk	pk	pkatt
0	2021-08-15	16:30	Premier League	Matchweek 1	Sun	Away	L	0.0	1.0	Tottenham	...	Anthony Taylor	Match Report	18.0	4.0	16.9	1.0	0.0	0.0
1	2021-08-21	15:00	Premier League	Matchweek 2	Sat	Home	W	5.0	0.0	Norwich City	...	Graham Scott	Match Report	16.0	4.0	17.3	1.0	0.0	0.0
2	2021-08-28	12:30	Premier League	Matchweek 3	Sat	Home	W	5.0	0.0	Arsenal	...	Martin Atkinson	Match Report	25.0	10.0	14.3	0.0	0.0	0.0
3	2021-09-11	15:00	Premier League	Matchweek 4	Sat	Away	W	1.0	0.0	Leicester City	...	Paul Tierney	Match Report	25.0	8.0	14.0	0.0	0.0	0.0
4	2021-09-18	15:00	Premier League	Matchweek 5	Sat	Home	D	0.0	0.0	Southampton	...	Jonathan Moss	Match Report	16.0	1.0	15.7	1.0	0.0	0.0

5 rows × 25 columns

In [119]:

```
matches.to_csv('matches2.csv', index=False)
```

In [120]:

```
data = pd.read_csv('matches2.csv')
```

In [121]:

```
data.head()
```

Out[121]:

	date	time	comp	round	day	venue	result	gf	ga	opponent	...	referee	match report	sh	sot	dist	fk	pk	pkatt
0	2021-08-15	16:30	Premier League	Matchweek 1	Sun	Away	L	0.0	1.0	Tottenham	...	Anthony Taylor	Match Report	18.0	4.0	16.9	1.0	0.0	0.0
1	2021-08-21	15:00	Premier League	Matchweek 2	Sat	Home	W	5.0	0.0	Norwich City	...	Graham Scott	Match Report	16.0	4.0	17.3	1.0	0.0	0.0
2	2021-08-28	12:30	Premier League	Matchweek 3	Sat	Home	W	5.0	0.0	Arsenal	...	Martin Atkinson	Match Report	25.0	10.0	14.3	0.0	0.0	0.0
3	2021-09-11	15:00	Premier League	Matchweek 4	Sat	Away	W	1.0	0.0	Leicester City	...	Paul Tierney	Match Report	25.0	8.0	14.0	0.0	0.0	0.0
4	2021-09-18	15:00	Premier League	Matchweek 5	Sat	Home	D	0.0	0.0	Southampton	...	Jonathan Moss	Match Report	16.0	1.0	15.7	1.0	0.0	0.0

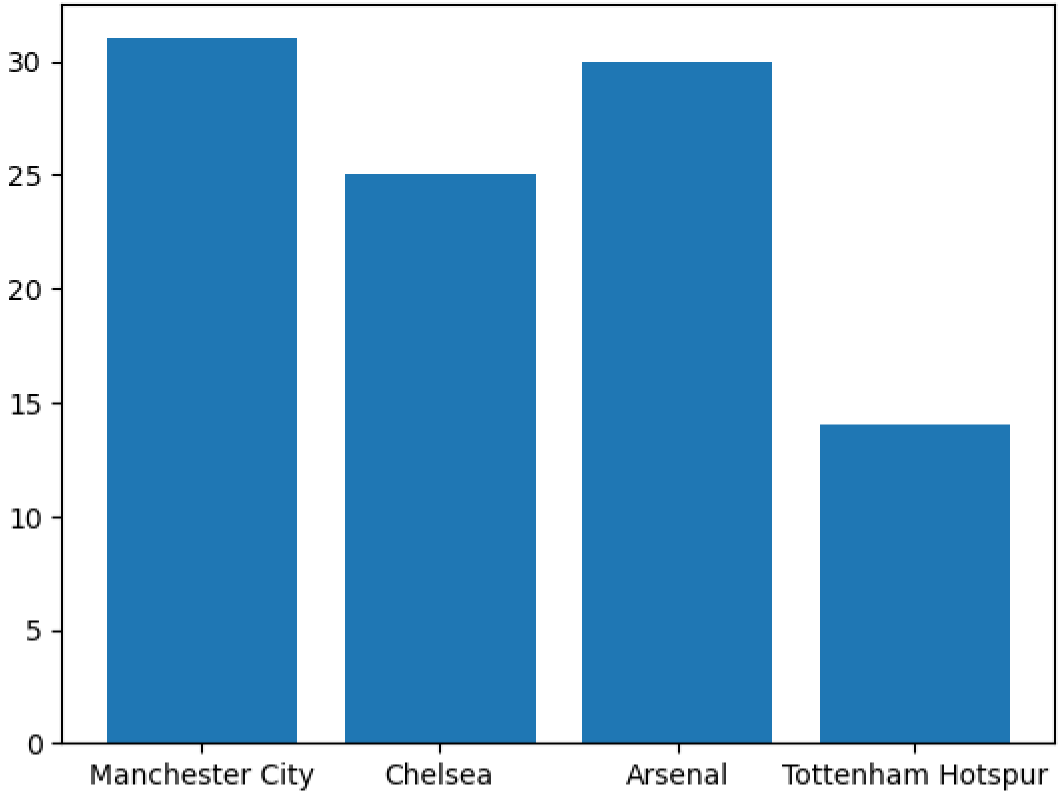
5 rows × 25 columns

In [122]:

```
import matplotlib.pyplot as plt
```

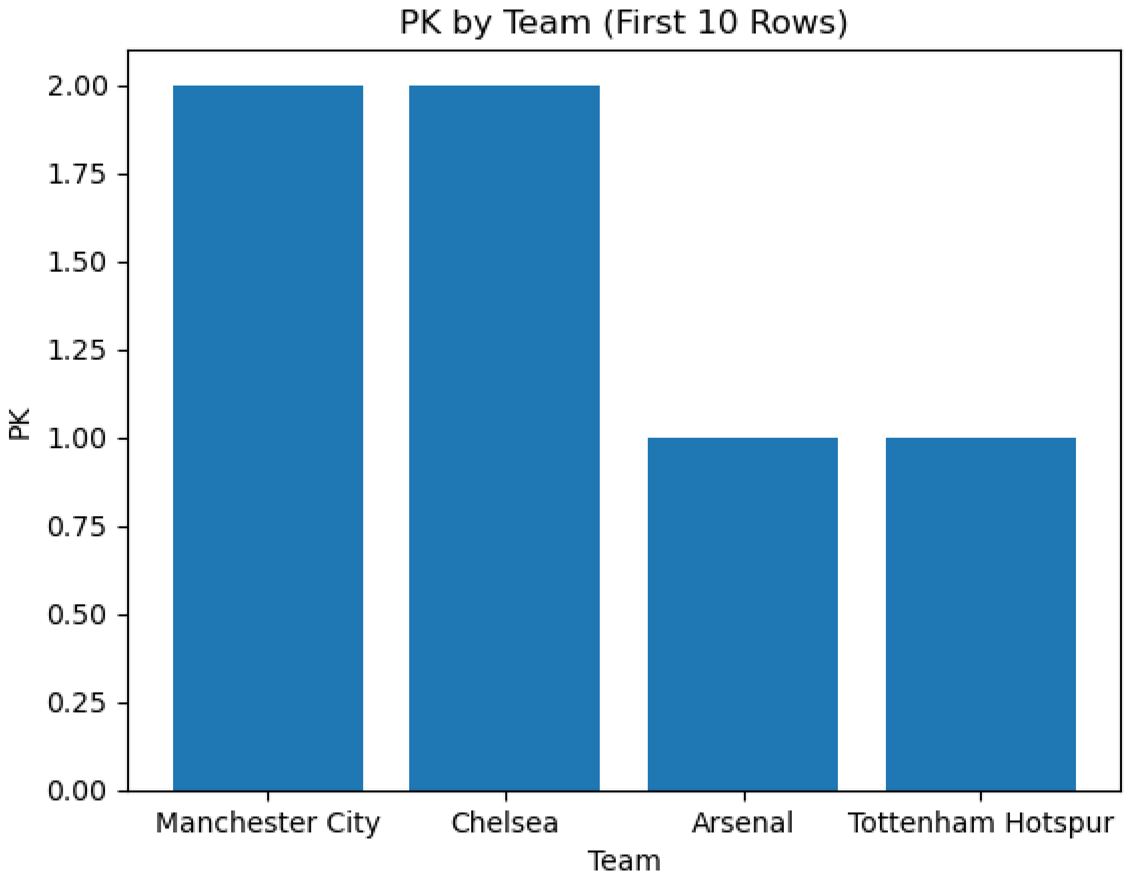
In [123]:

```
# team against sh
fig, ax = plt.subplots()
ax.bar(data['team'][:100], data['sh'][:100]);
```



In [124]:

```
# Plotting the bar plot
# team against pk
plt.bar(data['team'].head(100), data['pk'].head(100))
plt.xlabel('Team')
plt.ylabel('PK')
plt.title('PK by Team (First 10 Rows)')
plt.show()
```



In [125]:

```
data.date
```

Out[125]:

```
0       2021-08-15
1       2021-08-21
2       2021-08-28
3       2021-09-11
4       2021-09-18
           ...
1384    2021-05-02
1385    2021-05-08
1386    2021-05-16
1387    2021-05-19
1388    2021-05-23
Name: date, Length: 1389, dtype: object
```

Parsing date

In [126]:

```
data = pd.read_csv('matches2.csv',
                   low_memory=False,
                   parse_dates=['date'])
```

In [127]:

```
data.date
```

Out[127]:

```
0      2021-08-15
1      2021-08-21
2      2021-08-28
3      2021-09-11
4      2021-09-18
          ...
1384   2021-05-02
1385   2021-05-08
1386   2021-05-16
1387   2021-05-19
1388   2021-05-23
Name: date, Length: 1389, dtype: datetime64[ns]
```

In [128]:

```
data
```

Out[128]:

	date	time	comp	round	day	venue	result	gf	ga	opponent	...	referee	match report	sh	sot	dist	fk	pk	p
0	2021-08-15	16:30	Premier League	Matchweek 1	Sun	Away	L	0.0	1.0	Tottenham	...	Anthony Taylor	Match Report	18.0	4.0	16.9	1.0	0.0	
1	2021-08-21	15:00	Premier League	Matchweek 2	Sat	Home	W	5.0	0.0	Norwich City	...	Graham Scott	Match Report	16.0	4.0	17.3	1.0	0.0	
2	2021-08-28	12:30	Premier League	Matchweek 3	Sat	Home	W	5.0	0.0	Arsenal	...	Martin Atkinson	Match Report	25.0	10.0	14.3	0.0	0.0	
3	2021-09-11	15:00	Premier League	Matchweek 4	Sat	Away	W	1.0	0.0	Leicester City	...	Paul Tierney	Match Report	25.0	8.0	14.0	0.0	0.0	
4	2021-09-18	15:00	Premier League	Matchweek 5	Sat	Home	D	0.0	0.0	Southampton	...	Jonathan Moss	Match Report	16.0	1.0	15.7	1.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1384	2021-05-02	19:15	Premier League	Matchweek 34	Sun	Away	L	0.0	4.0	Tottenham	...	Andre Marriner	Match Report	8.0	1.0	17.4	0.0	0.0	
1385	2021-05-08	15:00	Premier League	Matchweek 35	Sat	Home	L	0.0	2.0	Crystal Palace	...	Simon Hooper	Match Report	7.0	0.0	11.4	1.0	0.0	
1386	2021-05-16	19:00	Premier League	Matchweek 36	Sun	Away	W	1.0	0.0	Everton	...	Jonathan Moss	Match Report	10.0	3.0	17.0	0.0	0.0	
1387	2021-05-19	18:00	Premier League	Matchweek 37	Wed	Away	L	0.0	1.0	Newcastle Utd	...	Robert Jones	Match Report	11.0	1.0	16.0	1.0	0.0	
1388	2021-05-23	16:00	Premier League	Matchweek 38	Sun	Home	W	1.0	0.0	Burnley	...	Kevin Friend	Match Report	12.0	3.0	17.0	0.0	0.0	

1389 rows × 25 columns

In [129]:

```
data.isna().sum()
```

Out[129]:

```
date            0
time            0
comp            0
round           0
day             0
venue           0
result          0
gf              0
ga              0
opponent        0
xg              0
xga             0
poss            0
captain         0
formation       0
referee         0
match report    0
sh              0
sot             0
```

In [130]:

```
data.dropna(subset=['dist'], inplace=True)

```

In [131]:
```
data.isna().sum()
```

Out[131]:
```
date            0
time            0
comp            0
round           0
day             0
venue           0
result          0
gf              0
ga              0
opponent        0
xg              0
xga             0
poss            0
captain         0
formation       0
referee         0
match report    0
sh              0
sot             0
dist            0
fk              0
pk              0
pkatt           0
season          0
team            0
dtype: int64
```

In [132]:
```
len(data)
```

Out[132]:
```
1388
```

Sort DataFrame by date

In [133]:

```
data.sort_values(by = ['date'], inplace = True, ascending = True)
data.date.head(100)
```

Out[133]:

```
1047   2020-09-12
1275   2020-09-12
705    2020-09-12
1161   2020-09-12
933    2020-09-12
          ...
1013   2020-10-23
938    2020-10-23
671    2020-10-24
824    2020-10-24
1128   2020-10-24
Name: date, Length: 100, dtype: datetime64[ns]
```

In [134]:

```
data_temp = data.copy()
```

In [135]:

```
data_temp.date
```

Out[135]:

```
1047   2020-09-12
1275   2020-09-12
705    2020-09-12
1161   2020-09-12
933    2020-09-12
          ...
530    2022-04-24
331    2022-04-24
399    2022-04-24
432    2022-04-25
497    2022-04-25
Name: date, Length: 1388, dtype: datetime64[ns]
```

In [136]:

```
data_temp
```

Out[136]:

	date	time	comp	round	day	venue	result	gf	ga	opponent	...	referee	match report	sh	sot	dist	fk	pk	pk
1047	2020-09-12	20:00	Premier League	Matchweek 1	Sat	Away	W	2.0	0.0	West Ham	...	Stuart Attwell	Match Report	16.0	3.0	16.2	1.0	0.0	
1275	2020-09-12	12:30	Premier League	Matchweek 1	Sat	Home	L	0.0	3.0	Arsenal	...	Chris Kavanagh	Match Report	5.0	2.0	26.0	0.0	0.0	
705	2020-09-12	17:30	Premier League	Matchweek 1	Sat	Home	W	4.0	3.0	Leeds United	...	Michael Oliver	Match Report	20.0	4.0	17.0	0.0	2.0	
1161	2020-09-12	15:00	Premier League	Matchweek 1	Sat	Away	L	0.0	1.0	Crystal Palace	...	Jonathan Moss	Match Report	9.0	5.0	15.6	2.0	0.0	
933	2020-09-12	17:30	Premier League	Matchweek 1	Sat	Away	L	3.0	4.0	Liverpool	...	Michael Oliver	Match Report	6.0	3.0	17.5	1.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
530	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Home	W	1.0	0.0	Wolves	...	Anthony Taylor	Match Report	13.0	5.0	18.8	0.0	0.0	
331	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Home	D	2.0	2.0	Southampton	...	Robert Jones	Match Report	8.0	5.0	11.2	0.0	0.0	
399	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Away	D	2.0	2.0	Brighton	...	Robert Jones	Match Report	18.0	5.0	19.4	1.0	0.0	
432	2022-04-25	20:00	Premier League	Matchweek 34	Mon	Home	D	0.0	0.0	Leeds United	...	Darren England	Match Report	17.0	7.0	13.8	0.0	0.0	
497	2022-04-25	20:00	Premier League	Matchweek 34	Mon	Away	D	0.0	0.0	Crystal Palace	...	Darren England	Match Report	9.0	2.0	16.5	0.0	0.0	

1388 rows × 25 columns

In [137]:

```
data_temp.describe()
```

Out[137]:

	gf	ga	xg	xga	poss	sh	sot	dist	fk	
count	1388.000000	1388.000000	1388.000000	1388.000000	1388.000000	1388.000000	1388.000000	1388.000000	1388.000000	1388
mean	1.335735	1.381124	1.304539	1.338617	49.713977	12.162104	4.043948	17.011527	0.456052	0
std	1.274662	1.291474	0.767425	0.789618	12.399196	5.260656	2.402282	2.988364	0.665516	0
min	0.000000	0.000000	0.000000	0.000000	18.000000	1.000000	0.000000	4.000000	0.000000	0
25%	0.000000	0.000000	0.700000	0.700000	40.000000	8.000000	2.000000	15.100000	0.000000	0
50%	1.000000	1.000000	1.200000	1.200000	50.000000	12.000000	4.000000	16.900000	0.000000	0
75%	2.000000	2.000000	1.800000	1.800000	59.000000	15.000000	5.000000	18.800000	1.000000	0
max	9.000000	9.000000	4.600000	5.000000	82.000000	31.000000	15.000000	34.900000	4.000000	3

In [138]:

```
data_temp[: 1].date.dt.year, data_temp[: 1].date.dt.month, data_temp[: 1].date.dt.day
```

Out[138]:

```
(1047    2020
 Name: date, dtype: int64,
 1047    9
 Name: date, dtype: int64,
 1047    12
 Name: date, dtype: int64)
```

In [139]:

```
data_temp[: 1].date
```

Out[139]:

```
1047   2020-09-12
Name: date, dtype: datetime64[ns]
```

In [140]:

```
data_temp['Year'] = data_temp.date.dt.year
data_temp['Month'] = data_temp.date.dt.month
data_temp['DayOfWeek'] = data_temp.date.dt.dayofweek
data_temp['DayOfYear'] = data_temp.date.dt.dayofyear
```

In [141]:

```
data_temp.head
```

Out[141]:

```
<bound method NDFrame.head of            date   time            comp         round  day venue result    g
f  \
1047 2020-09-12  20:00  Premier League   Matchweek 1  Sat  Away      W  2.0
1275 2020-09-12  12:30  Premier League   Matchweek 1  Sat  Home      L  0.0
705  2020-09-12  17:30  Premier League   Matchweek 1  Sat  Home      W  4.0
1161 2020-09-12  15:00  Premier League   Matchweek 1  Sat  Away      L  0.0
933  2020-09-12  17:30  Premier League   Matchweek 1  Sat  Away      L  3.0
...         ...    ...             ...           ...  ...   ...    ...  ...
530  2022-04-24  14:00  Premier League  Matchweek 34  Sun  Home      W  1.0
331  2022-04-24  14:00  Premier League  Matchweek 34  Sun  Home      D  2.0
399  2022-04-24  14:00  Premier League  Matchweek 34  Sun  Away      D  2.0
432  2022-04-25  20:00  Premier League  Matchweek 34  Mon  Home      D  0.0
497  2022-04-25  20:00  Premier League  Matchweek 34  Mon  Away      D  0.0

       ga         opponent  ...  dist   fk   pk pkatt season  \
1047  0.0         West Ham  ...  16.2  1.0  0.0   0.0   2021
1275  3.0          Arsenal  ...  26.0  0.0  0.0   0.0   2021
705   3.0     Leeds United  ...  17.0  0.0  2.0   2.0   2021
1161  1.0   Crystal Palace  ...  15.6  2.0  0.0   0.0   2021
933   4.0        Liverpool  ...  17.5  1.0  0.0   0.0   2021
...   ...              ...  ...   ...  ...  ...   ...    ...
530   0.0           Wolves  ...  18.8  0.0  0.0   0.0   2022
331   2.0      Southampton  ...  11.2  0.0  0.0   0.0   2022
399   2.0         Brighton  ...  19.4  1.0  0.0   0.0   2022
432   0.0     Leeds United  ...  13.8  0.0  0.0   0.0   2022
497   0.0   Crystal Palace  ...  16.5  0.0  0.0   0.0   2022

                          team  Year  Month  DayOfWeek  DayOfYear
1047          Newcastle United  2020      9          5        256
1275                    Fulham  2020      9          5        256
705                  Liverpool  2020      9          5        256
1161               Southampton  2020      9          5        256
933               Leeds United  2020      9          5        256
...                        ...   ...    ...        ...        ...
530                    Burnley  2022      4          6        114
331   Brighton and Hove Albion  2022      4          6        114
399                Southampton  2022      4          6        114
432             Crystal Palace  2022      4          0        115
497               Leeds United  2022      4          0        115

[1388 rows x 29 columns]>
```

In [142]:

```
# Now we've enrich our DataFrame with date time features, we can remove 'saledate'
data_temp.drop('date', axis = 1, inplace=True)
```

In [143]:

```
data_temp.result.value_counts()
```

Out[143]:

```
L    548
W    526
D    314
Name: result, dtype: int64
```

In [144]:

```
data_temp.isna().sum()
```

Out[144]:

```
time            0
comp            0
round           0
day             0
venue           0
result          0
gf              0
ga              0
opponent        0
xg              0
xga             0
poss            0
captain         0
formation       0
referee         0
match report    0
sh              0
sot             0
dist            0
fk              0
pk              0
pkatt           0
season          0
team            0
Year            0
Month           0
DayOfWeek       0
DayOfYear       0
dtype: int64
```

Feature Engineering

Convert string to categories

In [145]:

```
# find colums that contain strings
for label, content in data_temp.items():
    if pd.api.types.is_string_dtype(content):
        print(label)
```

```
time
comp
round
day
venue
result
opponent
captain
formation
referee
match report
team
```

In [146]:

```
# This will turn all of the string value into category values
for label, content in data_temp.items():
    if pd.api.types.is_string_dtype(content):
        data_temp[label] = content.astype('category').cat.as_ordered()

```

In [147]:

```
data_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1388 entries, 1047 to 497
Data columns (total 28 columns):
 #   Column        Non-Null Count  Dtype   
---  ------        --------------  -----   
 0   time          1388 non-null   category
 1   comp          1388 non-null   category
 2   round         1388 non-null   category
 3   day           1388 non-null   category
 4   venue         1388 non-null   category
 5   result        1388 non-null   category
 6   gf            1388 non-null   float64 
 7   ga            1388 non-null   float64 
 8   opponent      1388 non-null   category
 9   xg            1388 non-null   float64 
 10  xga           1388 non-null   float64 
 11  poss          1388 non-null   float64 
 12  captain       1388 non-null   category
 13  formation     1388 non-null   category
 14  referee       1388 non-null   category
 15  match report  1388 non-null   category
 16  sh            1388 non-null   float64 
 17  sot           1388 non-null   float64 
 18  dist          1388 non-null   float64 
 19  fk            1388 non-null   float64 
 20  pk            1388 non-null   float64 
 21  pkatt         1388 non-null   float64 
 22  season        1388 non-null   int64   
 23  team          1388 non-null   category
 24  Year          1388 non-null   int64   
 25  Month         1388 non-null   int64   
 26  DayOfWeek     1388 non-null   int64   
 27  DayOfYear     1388 non-null   int64   
dtypes: category(12), float64(11), int64(5)
memory usage: 209.1 KB
```

In [148]:

```python
for label, content in data_temp.items():
    if pd.api.types.is_numeric_dtype(content):
        print(label)
```

```
gf
ga
xg
xga
poss
sh
sot
dist
fk
pk
pkatt
season
Year
Month
DayOfWeek
DayOfYear
```

In [149]:

```python
# checking for null value
for label, content in data_temp.items():
    if pd.api.types.is_numeric_dtype(content):
        if pd.isnull(content).sum():
            print(label)

```

Turn categories into numbers

In [150]:

```python
for label, content in data_temp.items():
    if not pd.api.types.is_numeric_dtype(content):
        data_temp[label] = pd.Categorical(content).codes + 1
```

In [151]:

```
data_temp
```

Out[151]:

	time	comp	round	day	venue	result	gf	ga	opponent	xg	...	dist	fk	pk	pkatt	season	team	Year	Month	DayOfWee
1047	17	1	1	3	1	3	2.0	0.0	22	1.5	...	16.2	1.0	0.0	0.0	2021	15	2020	9	
1275	2	1	1	3	2	2	0.0	3.0	1	0.2	...	26.0	0.0	0.0	0.0	2021	9	2020	9	
705	10	1	1	3	2	3	4.0	3.0	10	3.3	...	17.0	0.0	2.0	2.0	2021	12	2020	9	
1161	7	1	1	3	1	2	0.0	1.0	7	0.8	...	15.6	2.0	0.0	0.0	2021	18	2020	9	
933	10	1	1	3	1	2	3.0	4.0	12	0.6	...	17.5	1.0	0.0	0.0	2021	10	2020	9	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
530	4	1	28	4	2	3	1.0	0.0	23	1.0	...	18.8	0.0	0.0	0.0	2022	5	2022	4	
331	4	1	28	4	2	1	2.0	2.0	18	1.4	...	11.2	0.0	0.0	0.0	2022	4	2022	4	
399	4	1	28	4	1	1	2.0	2.0	4	0.9	...	19.4	1.0	0.0	0.0	2022	18	2022	4	
432	17	1	28	2	2	1	0.0	0.0	10	2.0	...	13.8	0.0	0.0	0.0	2022	7	2022	4	
497	17	1	28	2	1	1	0.0	0.0	7	0.4	...	16.5	0.0	0.0	0.0	2022	10	2022	4	

1388 rows × 28 columns

Building our model

splitting data into training and test sets

In [189]:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

In [190]:

```
train_df = data_temp[data_temp.index < 2022]
test_df = data_temp[data_temp.index > 2022]
# Model training
x = train_df.drop(columns=['result'])
y = train_df['result']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

```

In [191]:

```
#  Model selection
random_state = 42  # Set the random state for reproducibility
model = RandomForestClassifier(random_state=random_state)
```

In [192]:

```
x_train.shape, y_train.shape
```

Out[192]:

```
((1110, 27), (1110,))
```

In [193]:

```
model.fit(x_train, y_train)
```

Out[193]:

```
RandomForestClassifier(random_state=42)
```

In [194]:

```
y_pred = model.predict(x_test)
y_pred
```

Out[194]:

```
array([2, 3, 3, 3, 3, 2, 1, 3, 3, 3, 2, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 2,
       2, 1, 3, 3, 3, 3, 2, 2, 2, 2, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 1, 2,
       1, 3, 2, 2, 2, 1, 2, 3, 2, 3, 3, 1, 3, 1, 3, 1, 2, 3, 3, 1, 2, 3,
       2, 2, 1, 1, 3, 1, 2, 2, 2, 2, 2, 2, 3, 2, 1, 1, 1, 2, 3, 1, 2, 2,
       2, 3, 1, 3, 3, 3, 2, 2, 2, 3, 1, 2, 2, 3, 3, 2, 3, 2, 2, 3, 2, 2,
       3, 3, 2, 3, 3, 2, 3, 2, 3, 3, 3, 3, 2, 2, 1, 3, 1, 2, 1, 1, 1, 3,
       2, 1, 2, 1, 3, 3, 3, 1, 2, 3, 3, 3, 3, 3, 1, 2, 3, 2, 2, 3, 1, 3,
       2, 3, 3, 1, 3, 3, 3, 3, 2, 2, 2, 3, 3, 1, 3, 2, 2, 3, 2, 3, 3, 1,
       1, 1, 2, 3, 3, 2, 2, 2, 3, 2, 1, 3, 2, 3, 2, 2, 3, 1, 1, 1, 2, 2,
       1, 1, 3, 3, 1, 3, 2, 3, 3, 2, 2, 3, 1, 3, 2, 3, 2, 2, 3, 3, 2, 2,
       3, 1, 3, 2, 3, 3, 2, 3, 3, 3, 1, 2, 2, 2, 1, 3, 3, 2, 1, 2, 2, 2,
       1, 1, 1, 2, 3, 2, 2, 3, 3, 3, 3, 3, 2, 2, 2, 2, 3, 2, 3, 1, 2, 3,
       3, 2, 1, 3, 1, 3, 3, 3, 2, 3, 3, 3, 2, 3], dtype=int8)
```

In [195]:

```
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Accuracy: 0.960431654676259
```

In [196]:

```
label_mapping = {1: 'Draw', 2: 'Loss', 3: 'Win'}

# Replace numeric predictions with labels
y_pred_labels = [label_mapping[pred] for pred in y_pred]

# Print the predicted labels
print(y_pred_labels)

```

```
['Loss', 'Win', 'Win', 'Win', 'Win', 'Loss', 'Draw', 'Win', 'Win', 'Win', 'Loss', 'Win', 'Win', 'Loss',
'Win', 'Loss', 'Loss', 'Loss', 'Loss', 'Loss', 'Loss', 'Loss', 'Loss', 'Draw', 'Win', 'Win', 'Win', 'Wi
n', 'Loss', 'Loss', 'Loss', 'Loss', 'Draw', 'Loss', 'Win', 'Win', 'Win', 'Win', 'Win', 'Win', 'Win', 'Wi
n', 'Draw', 'Loss', 'Draw', 'Win', 'Loss', 'Loss', 'Loss', 'Draw', 'Loss', 'Win', 'Loss', 'Win', 'Win',
'Draw', 'Win', 'Draw', 'Win', 'Draw', 'Loss', 'Win', 'Win', 'Draw', 'Loss', 'Win', 'Loss', 'Loss', 'Dra
w', 'Draw', 'Win', 'Draw', 'Loss', 'Loss', 'Loss', 'Loss', 'Loss', 'Loss', 'Win', 'Loss', 'Draw', 'Dra
w', 'Draw', 'Loss', 'Win', 'Draw', 'Loss', 'Loss', 'Loss', 'Win', 'Draw', 'Win', 'Win', 'Win', 'Loss',
'Loss', 'Loss', 'Win', 'Draw', 'Loss', 'Loss', 'Win', 'Win', 'Loss', 'Win', 'Loss', 'Loss', 'Win', 'Los
s', 'Loss', 'Win', 'Win', 'Loss', 'Win', 'Win', 'Loss', 'Win', 'Loss', 'Win', 'Win', 'Win', 'Win', 'Los
s', 'Loss', 'Draw', 'Win', 'Draw', 'Loss', 'Draw', 'Draw', 'Draw', 'Win', 'Loss', 'Draw', 'Loss', 'Dra
w', 'Win', 'Win', 'Win', 'Draw', 'Loss', 'Win', 'Win', 'Win', 'Win', 'Win', 'Draw', 'Loss', 'Win', 'Los
s', 'Loss', 'Win', 'Draw', 'Win', 'Loss', 'Win', 'Win', 'Draw', 'Win', 'Win', 'Win', 'Win', 'Loss', 'Los
s', 'Loss', 'Win', 'Win', 'Draw', 'Win', 'Loss', 'Loss', 'Win', 'Loss', 'Win', 'Win', 'Draw', 'Draw', 'D
raw', 'Loss', 'Win', 'Win', 'Loss', 'Loss', 'Loss', 'Win', 'Loss', 'Draw', 'Win', 'Loss', 'Win', 'Loss',
'Loss', 'Win', 'Draw', 'Draw', 'Draw', 'Loss', 'Loss', 'Draw', 'Draw', 'Win', 'Win', 'Draw', 'Win', 'Los
s', 'Win', 'Win', 'Loss', 'Loss', 'Win', 'Draw', 'Win', 'Loss', 'Win', 'Loss', 'Loss', 'Win', 'Win', 'Lo
ss', 'Loss', 'Win', 'Draw', 'Win', 'Loss', 'Win', 'Win', 'Loss', 'Win', 'Win', 'Win', 'Draw', 'Loss', 'L
oss', 'Loss', 'Draw', 'Win', 'Win', 'Loss', 'Draw', 'Loss', 'Loss', 'Loss', 'Draw', 'Draw', 'Draw', 'Los
s', 'Win', 'Loss', 'Loss', 'Win', 'Win', 'Win', 'Win', 'Win', 'Loss', 'Loss', 'Loss', 'Loss', 'Win', 'Lo
ss', 'Win', 'Draw', 'Loss', 'Win', 'Win', 'Loss', 'Draw', 'Win', 'Draw', 'Win', 'Win', 'Win', 'Loss', 'W
in', 'Win', 'Win', 'Loss', 'Win']
```

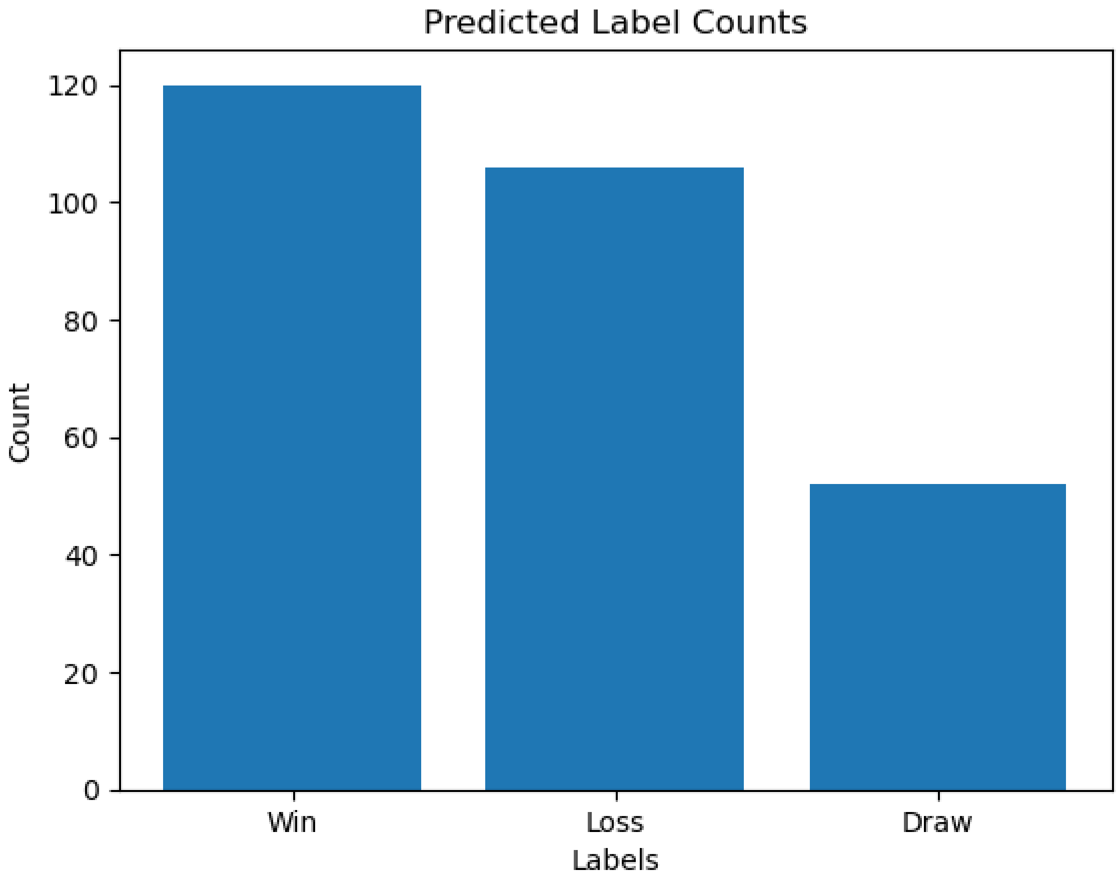
In [197]:

```
# Convert numeric predictions to labels
y_pred_labels = [label_mapping[pred] for pred in y_pred]

# Create a pandas Series with the predicted labels
y_pred_series = pd.Series(y_pred_labels)

# Count the occurrences of each predicted label
label_counts = y_pred_series.value_counts()

# Print the label counts
print(label_counts)
```

```
Win     120
Loss    106
Draw     52
dtype: int64
```

In [198]:

```
# Plot the bar chart
plt.bar(label_counts.index, label_counts.values)
plt.xlabel('Labels')
plt.ylabel('Count')
plt.title('Predicted Label Counts')
plt.show()
```



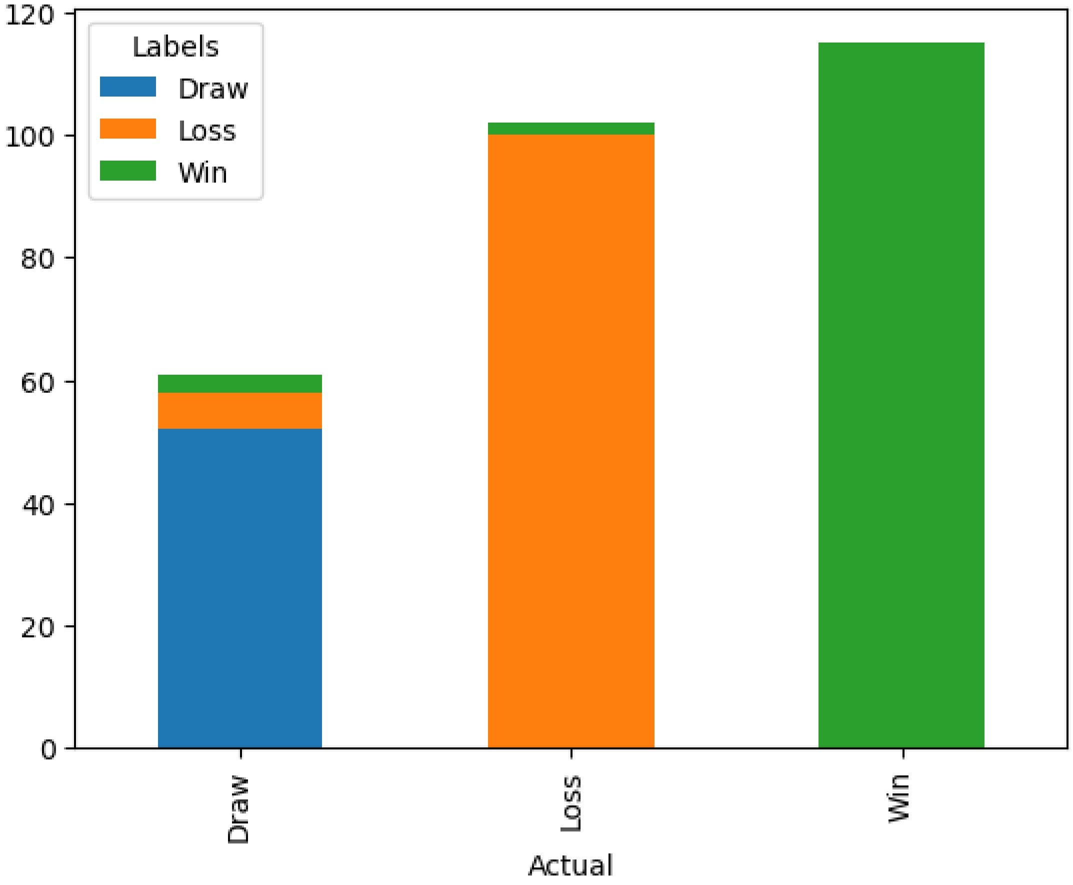
In [199]:

```python
import pandas as pd

# Define the label mapping
label_mapping = {1: 'Draw', 2: 'Loss', 3: 'Win'}

# Replace numeric predictions with labels
y_pred_labels = [label_mapping[pred] for pred in y_pred]

# Replace numeric true labels with labels
y_test_labels = [label_mapping[true_label] for true_label in y_test]  # Replace y_test with your true

# Create a pandas DataFrame with the actual and predicted labels
df = pd.DataFrame({'Actual': y_test_labels, 'Predicted': y_pred_labels})

# Create a cross-tabulation of the actual and predicted labels
cross_tab = pd.crosstab(df['Actual'], df['Predicted'])

# Print the cross-tabulation
print(cross_tab)

```

```
Predicted  Draw  Loss  Win
Actual
Draw         52     6    3
Loss          0   100    2
Win           0     0  115
```

In [200]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Define the label mapping
label_mapping = {1: 'Draw', 2: 'Loss', 3: 'Win'}

# Replace numeric predictions with labels
y_pred_labels = [label_mapping[pred] for pred in y_pred]

# Replace numeric true labels with labels
y_test_labels = [label_mapping[true_label] for true_label in y_test]  # Replace y_test with your true

# Create a pandas DataFrame with the actual and predicted labels
df = pd.DataFrame({'Actual': y_test_labels, 'Predicted': y_pred_labels})

# Create a cross-tabulation of the actual and predicted labels
cross_tab = pd.crosstab(df['Actual'], df['Predicted'])

# Plot the cross-tabulation
cross_tab.plot(kind='bar', stacked=True)

# Add legend
plt.legend(title='Labels')

# Display the plot
plt.show()

```



Hyperparameter turning with RandomizedSearchCV

In [201]:

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint

# Define the parameter distribution for RandomizedSearchCV
param_dist = {
    'n_estimators': randint(100, 1000),  # Number of trees in the forest
    'max_depth': randint(1, 20),  # Maximum depth of each tree
    'max_features': ['auto', 'sqrt'],  # Number of features to consider at each split
    'min_samples_split': randint(2, 10),  # Minimum number of samples required to split an internal n
    'min_samples_leaf': randint(1, 10)  # Minimum number of samples required to be at a leaf node
}

```

In [202]:

```python

# Create a RandomForestClassifier instance
model = RandomForestClassifier()

# Create a RandomizedSearchCV instance
random_search = RandomizedSearchCV(
    estimator=model,
    param_distributions=param_dist,
    n_iter=10,  # Number of parameter settings that are sampled
    cv=5,  # Number of cross-validation folds
    random_state=42
)

# Perform the random search to find the best hyperparameters
random_search.fit(x_train, y_train)

# Print the best hyperparameters and the corresponding accuracy
print("Best Hyperparameters: ", random_search.best_params_)
print("Best Accuracy: ", random_search.best_score_)

```

```
Best Hyperparameters:  {'max_depth': 9, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_spli
t': 5, 'n_estimators': 700}
Best Accuracy:  0.963063063063063
```

In [203]:

```python
# Create a RandomForestClassifier instance with the best hyperparameters
best_rf_classifier = RandomForestClassifier(
    n_estimators=700,
    max_depth=9,
    max_features='auto',
    min_samples_leaf=2,
    min_samples_split=5
)

# Fit the classifier with the best hyperparameters to the training data
best_rf_classifier.fit(x_train, y_train)

# Evaluate the performance on the test set
accuracy = best_rf_classifier.score(x_test, y_test)
print("Test Accuracy with Best Hyperparameters: ", accuracy)

```

```
Test Accuracy with Best Hyperparameters:  0.9568345323741008
```

In [204]:

```python
import joblib

# Save the trained model to a file
joblib.dump(best_rf_classifier, 'best_rf_model.pkl')

```

Out[204]:

```
['best_rf_model.pkl']
```

In [205]:

```python
# Load the saved model from file
loaded_model = joblib.load('best_rf_model.pkl')

# Use the loaded model for predictions
predictions = loaded_model.predict(x_test)

```

In [206]:
```
predictions
```

Out[206]:
```
array([2, 3, 3, 3, 3, 2, 1, 3, 3, 3, 2, 3, 3, 1, 3, 2, 2, 2, 2, 2, 2, 2,
       2, 1, 3, 3, 3, 3, 2, 2, 2, 2, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 1, 2,
       1, 3, 2, 2, 2, 1, 2, 3, 2, 3, 3, 1, 3, 1, 3, 1, 2, 3, 3, 1, 2, 3,
       2, 2, 1, 1, 3, 1, 2, 2, 2, 2, 2, 2, 3, 2, 3, 1, 1, 2, 3, 1, 2, 2,
       2, 3, 2, 3, 3, 3, 2, 2, 2, 3, 1, 2, 2, 3, 3, 2, 3, 2, 2, 3, 2, 2,
       3, 3, 2, 3, 3, 2, 3, 2, 3, 3, 3, 3, 2, 2, 1, 3, 1, 2, 1, 1, 1, 3,
       2, 1, 2, 1, 3, 3, 3, 1, 2, 3, 3, 3, 3, 3, 1, 2, 3, 2, 2, 3, 1, 3,
       2, 3, 3, 1, 3, 3, 3, 3, 2, 2, 2, 3, 3, 1, 3, 2, 2, 3, 2, 3, 3, 1,
       1, 1, 2, 3, 3, 2, 2, 2, 3, 2, 1, 3, 2, 3, 2, 2, 3, 1, 1, 1, 2, 2,
       1, 1, 3, 3, 1, 3, 2, 3, 3, 2, 2, 3, 1, 3, 2, 3, 2, 2, 3, 3, 2, 2,
       3, 1, 3, 2, 3, 3, 2, 3, 3, 3, 1, 2, 2, 2, 1, 3, 3, 2, 1, 2, 2, 2,
       1, 1, 1, 2, 3, 2, 2, 3, 3, 3, 3, 3, 2, 2, 2, 2, 3, 3, 3, 1, 2, 3,
       3, 2, 1, 3, 1, 3, 3, 3, 2, 3, 3, 3, 2, 3], dtype=int8)
```

In [212]:
```
df_temp = pd.DataFrame(data_temp)
df_predictions = pd.DataFrame({'Prediction': predictions})

```

In [213]:

```
df_predictions
```

Out[213]:

	Prediction
0	2
1	3
2	3
3	3
4	3
...	...
273	3
274	3
275	3
276	2
277	3

278 rows × 1 columns

In [214]:
```
data
```

Out[214]:

	date	time	comp	round	day	venue	result	gf	ga	opponent	...	referee	match report	sh	sot	dist	fk	pk	pk
1047	2020-09-12	20:00	Premier League	Matchweek 1	Sat	Away	W	2.0	0.0	West Ham	...	Stuart Attwell	Match Report	16.0	3.0	16.2	1.0	0.0	
1275	2020-09-12	12:30	Premier League	Matchweek 1	Sat	Home	L	0.0	3.0	Arsenal	...	Chris Kavanagh	Match Report	5.0	2.0	26.0	0.0	0.0	
705	2020-09-12	17:30	Premier League	Matchweek 1	Sat	Home	W	4.0	3.0	Leeds United	...	Michael Oliver	Match Report	20.0	4.0	17.0	0.0	2.0	
1161	2020-09-12	15:00	Premier League	Matchweek 1	Sat	Away	L	0.0	1.0	Crystal Palace	...	Jonathan Moss	Match Report	9.0	5.0	15.6	2.0	0.0	
933	2020-09-12	17:30	Premier League	Matchweek 1	Sat	Away	L	3.0	4.0	Liverpool	...	Michael Oliver	Match Report	6.0	3.0	17.5	1.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
530	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Home	W	1.0	0.0	Wolves	...	Anthony Taylor	Match Report	13.0	5.0	18.8	0.0	0.0	
331	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Home	D	2.0	2.0	Southampton	...	Robert Jones	Match Report	8.0	5.0	11.2	0.0	0.0	
399	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Away	D	2.0	2.0	Brighton	...	Robert Jones	Match Report	18.0	5.0	19.4	1.0	0.0	
432	2022-04-25	20:00	Premier League	Matchweek 34	Mon	Home	D	0.0	0.0	Leeds United	...	Darren England	Match Report	17.0	7.0	13.8	0.0	0.0	
497	2022-04-25	20:00	Premier League	Matchweek 34	Mon	Away	D	0.0	0.0	Crystal Palace	...	Darren England	Match Report	9.0	2.0	16.5	0.0	0.0	

1388 rows × 25 columns

In [215]:
```
data.dropna(subset=['dist'], inplace=True)
```

In [216]:

```
data
```

Out[216]:

	date	time	comp	round	day	venue	result	gf	ga	opponent	...	referee	match report	sh	sot	dist	fk	pk	pk
1047	2020-09-12	20:00	Premier League	Matchweek 1	Sat	Away	W	2.0	0.0	West Ham	...	Stuart Attwell	Match Report	16.0	3.0	16.2	1.0	0.0	
1275	2020-09-12	12:30	Premier League	Matchweek 1	Sat	Home	L	0.0	3.0	Arsenal	...	Chris Kavanagh	Match Report	5.0	2.0	26.0	0.0	0.0	
705	2020-09-12	17:30	Premier League	Matchweek 1	Sat	Home	W	4.0	3.0	Leeds United	...	Michael Oliver	Match Report	20.0	4.0	17.0	0.0	2.0	
1161	2020-09-12	15:00	Premier League	Matchweek 1	Sat	Away	L	0.0	1.0	Crystal Palace	...	Jonathan Moss	Match Report	9.0	5.0	15.6	2.0	0.0	
933	2020-09-12	17:30	Premier League	Matchweek 1	Sat	Away	L	3.0	4.0	Liverpool	...	Michael Oliver	Match Report	6.0	3.0	17.5	1.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
530	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Home	W	1.0	0.0	Wolves	...	Anthony Taylor	Match Report	13.0	5.0	18.8	0.0	0.0	
331	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Home	D	2.0	2.0	Southampton	...	Robert Jones	Match Report	8.0	5.0	11.2	0.0	0.0	
399	2022-04-24	14:00	Premier League	Matchweek 34	Sun	Away	D	2.0	2.0	Brighton	...	Robert Jones	Match Report	18.0	5.0	19.4	1.0	0.0	
432	2022-04-25	20:00	Premier League	Matchweek 34	Mon	Home	D	0.0	0.0	Leeds United	...	Darren England	Match Report	17.0	7.0	13.8	0.0	0.0	
497	2022-04-25	20:00	Premier League	Matchweek 34	Mon	Away	D	0.0	0.0	Crystal Palace	...	Darren England	Match Report	9.0	2.0	16.5	0.0	0.0	

1388 rows × 25 columns

In [217]:

```
from sklearn.model_selection import train_test_split

```

In [218]:

```python
train_df = data[data.index < 2022]
test_df = data[data.index > 2022]
# Model training
x = train_df.drop(columns=['result'])
y = train_df['result']

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

In [219]:

```python
import pandas as pd

# Reset the indexes of the DataFrames
df_predictions_reset = df_predictions.reset_index(drop=True)
x_test_reset = x_test.reset_index(drop=True)

# Merge the DataFrames
merged_df = pd.concat([x_test_reset, df_predictions_reset], axis=1)

```

In [220]:

```python
columns_to_drop = ['comp', 'round', 'day', 'venue', 'match report', 'sh', 'sot', 'dist', 'fk', 'pk',

merged_df = merged_df.drop(columns=columns_to_drop)
```

In [221]:

```
merged_df.T
```

Out[221]:

	0	1	2	3	4	5	6	7	8	9	...	2
date	2021-04-25 00:00:00	2020-12-06 00:00:00	2021-02-04 00:00:00	2021-05-08 00:00:00	2022-03-10 00:00:00	2020-12-16 00:00:00	2021-12-01 00:00:00	2021-03-04 00:00:00	2022-03-01 00:00:00	2022-04-24 00:00:00	...	2020-12 00:00
time	19:00	14:15	20:00	12:30	19:45	20:00	19:30	20:15	19:45	14:00	...	12
gf	2.0	2.0	1.0	3.0	3.0	1.0	1.0	1.0	2.0	1.0	...	
ga	2.0	1.0	0.0	1.0	0.0	2.0	1.0	0.0	0.0	0.0	...	
opponent	Aston Villa	Sheffield Utd	Tottenham	Tottenham	Leeds United	Liverpool	Brighton	Liverpool	Burnley	West Ham	...	Leices (
xg	1.4	1.5	2.2	2.6	1.4	1.3	1.6	1.0	1.5	2.8	...	
xga	2.3	0.3	0.3	1.0	0.2	1.2	1.1	0.3	1.0	0.5	...	
poss	30.0	69.0	58.0	52.0	49.0	25.0	35.0	45.0	55.0	66.0	...	4
captain	Kyle Bartley	Kasper Schmeichel	César Azpilicueta	Luke Ayling	Tyrone Mings	Hugo Lloris	Declan Rice	César Azpilicueta	Kasper Schmeichel	César Azpilicueta	...	Ha Magu
formation	4-1-4-1	3-4-3	3-4-3	4-1-4-1	4-4-2◆	4-4-2	4-2-3-1	3-4-3	4-3-3	3-4-1-2	...	4-2-
referee	Stuart Attwell	Stuart Attwell	Andre Marriner	Michael Oliver	Simon Hooper	Anthony Taylor	Chris Kavanagh	Martin Atkinson	Chris Kavanagh	Michael Oliver	...	Mike De
team	West Bromwich Albion	Leicester City	Chelsea	Leeds United	Aston Villa	Tottenham Hotspur	West Ham United	Chelsea	Leicester City	Chelsea	...	Manches Uni
Prediction	2	3	3	3	3	2	1	3	3	3	...	

13 rows × 278 columns

In [222]:

```
# here 3 = win, 2 = loss, 1 = Draw
```